IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	)	
	)	
Guy L. STEELE, Jr.	)	Group Art Unit: 2193
	)	
Application No.: 10/035,747	)	Examiner: Mai, Tan V
	)	
Filed: December 28, 2001	)	
	)	
For: FLOATING POINT SYSTEM THAT REPRESENTS STATUS FLAG INFORMATION WITHIN A FLOATING POINT OPERAND	)	Confirmation No.: 3637

Attention: Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF UNDER BOARD RULE § 41.37

In support of the Notice of Appeal filed September 30, 2005, and further to Board Rule 41.37, Appellant presents this brief and enclose herewith a check for the fee of $500.00 required under 37 C.F.R. § 1.17(c).

This Appeal Brief is being filed concurrently with a Petition for an Extension of Time for four months and the appropriate fee.

This Appeal responds to the Notice of Panel Decision from Pre-Appeal Brief Review mailed on November 8, 2005 and the Final Office Action mailed on June 6, 2005, which rejected claims 1-5, 21-31, and 33-54 under 35 U.S.C. § 102(b) and rejected claims 6-20 and 32 under 35 U.S.C. § 103(a).

If any additional fees are required or if the enclosed payment is insufficient,

Appellant requests that the required fees be charged to Deposit Account No. 06-0916.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

Sun Microsystems, Inc. is the real party in interest, as indicated by the assignment in its name, recorded at Reel 012443, Frame 0596 on December 28, 2001.

II. RELATED APPEALS AND INTERFERENCES

In accordance with 37 C.F.R. § 41.37(c)(1)(ii), Appellant advises the Board of Patent Appeals and Interferences (the "Board") of the following pending appeals, which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the instant appeal:

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,579, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,595, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,584, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,587, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,647, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,580, in which an Appeal Brief was filed concurrently herewith.

III. STATUS OF CLAIMS

Claims 1-54 remain pending and under current examination.

Claims 1-5, 21-31, and 33-54 have been finally rejected by the Examiner under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,995,991 to Huang et al. ("*Huang*"); claims 1-5, 21-31, and 33-54 have been finally rejected by the Examiner under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,009,551 to Lynch et al. ("*Lynch*"); claims 6-20 and 32 have been finally rejected by the Examiner under 35 U.S.C. § 103(a) as being unpatentable over *Huang*; and claims 6-20 and 32 have been finally rejected by the Examiner under 35 U.S.C. § 103(a) as being unpatentable over *Lynch*. Appellant appeals the rejections of those claims. The attached Appendix contains a clean copy of the claims involved in the appeal, claims 1-54.

IV. STATUS OF AMENDMENTS

All amendments have been entered. No amendments under 37 C.F.R. § 1.116 have been filed.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claims 1 and 16 of this application recite a floating point operand data structure. *Specification*, p. 15, paragraph 040. Independent claims 21 and 27 recite a floating point system for performing at least one floating point operation. *Id.* at pp. 16-17, paragraph 43. Independent claims 33, 40, 44, and 51 recite methods and computer readable media for encoding a floating point operand with status information without maintaining the status information in a floating point status register. *Id.* at pp. 17-18, paragraphs 44, 46, and 47.

Digital electronic devices, such as digital computers, calculators and other devices, perform arithmetic calculations on values in integer, or "fixed point," format, in

fractional, or "floating point" format, or both. *Specification*, p. 2, paragraph 002. Institute of Electrical and Electronic Engineers (IEEE) Standard 754, (hereinafter "IEEE Std. 754") published in 1985 and adopted by the American National Standards Institute (ANSI), defines several standard formats for expressing values in floating point format and a number of aspects regarding behavior of computation in connection therewith. *Id.*

In prior art devices that perform floating point computations, floating point status information generated by the computation is represented as flags in a floating point status register. *Id.* at p. 12, paragraph 031.

However, the modes (*e.g.*, the rounding modes and traps enabled/disabled mode), flags (*e.g.*, flags representing the status information), and traps that are required to implement IEEE Std. 754 introduce implicit serialization issues. *Id.* at p. 13, paragraph 034. Implicit serialization is essentially the need for serial control of access (read/write) to and from globally used registers, such as a floating point status register. *Id.* The potential for implicit serialization makes the Standard difficult to implement coherently and efficiently in today's superscalar and parallel processing architectures without loss of performance. *Id.*

Moreover, the implicit side effects of a procedure that can change the flags or modes can make it very difficult for compilers to perform optimizations on floating point code. *Id.* at pp. 13-14, paragraph 035. As a result, compilers for most languages usually assume that every procedure call is an optimization barrier in order to be safe. *Id.* This unfortunately may lead to further loss of performance. *Id.*

The claimed invention addresses these and other problems of prior art floating point computational systems. *Id.* at p. 14, paragraph 039. Since the floating point

status information relating to a floating point operation is in the result generated for the operation, implicit serialization required by maintaining the floating point status information separate and apart therefrom may be obviated. *Id.* at pp. 14-15, paragraph 039.

The invention, as recited by independent claim 1, relates to a floating point operand data structure (Fig. 4) used in floating point computations and processing within a processing device (Fig. 3). *Id.* at p. 15, paragraph 040. The operand data structure may include a first portion of the data structure having floating point operand data (Fig. 4). The operand data structure may also include a second portion of the data structure having embedded status information associated with at least one status condition of the floating point operand data (Fig. 4). *Id.* at p. 22, paragraph 063; p. 15, paragraph 040.

The invention, as recited by independent claim 16, relates to a floating point operand data structure (Fig. 4) used by a processing device (Fig. 3) when performing floating point operations. *Id.* at p. 15, paragraph 040. The operand data structure may include a first data field having sign information associated with the floating point operand and a second data field having exponent information associated with the floating point operand (Fig. 4). *Id.* The operand data structure may also include a third data field having fractional information associated with the floating point operand, wherein at least one of the first data field, the second data field and the third data field further includes embedded status information associated with at least one operand status condition (Fig. 4). *Id.* at pp. 15-16, paragraph 041.

The invention, as recited by independent claim 21, relates to a floating point system associated with a processing device for performing at least one floating point operation on a floating point operand (Fig. 3). The system may include an operand memory storage device for maintaining the floating point operand (Fig. 3, 51). The system may also include a control unit (Fig. 3., 50) in communication with the operand memory storage device (Fig. 3, 51), the control unit receiving at least one floating point instruction associated with the at least one floating point operation and generating at least one control signal related to the at least one floating point operation. *Id.* at p. 21, paragraph 060. Further, the system may include a first functional processing unit (Fig. 3, 41-47) in communication with the operand memory storage device (Fig. 3, 51) and the control unit (Fig. 3, 50), the first functional processing unit capable processing the floating point operand and storing status information within the processed floating point operand. *Id.* at pp. 16-17, paragraph 043; pp. 21-22, paragraphs 061-063.

The invention, as recited by independent claim 27, relates to a floating point processing system for performing at least one floating point operation on a floating point operand (Fig. 3). The system may include an operand memory register (Fig. 3, 51) for maintaining the floating point operand. The system may also include a functional processing unit (Fig. 3, 41-47) in communication with the operand memory register (Fig. 3, 51), the functional processing unit being operative to receive the floating point operand from the operand memory register, process the floating point operand to determine status information related to the processed floating point operand, and embed the status information within the processed floating point operand. *Id.* at pp. 16-17, paragraph 043; pp. 21-22, paragraph 061-063.

The invention, as recited by independent claim 33, relates to a method of encoding a floating point operand with status information without maintaining the status information in a floating point status register. *Id.* at p. 17, paragraph 044; p. 22, paragraph 063. The method may include determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation. *Id.* at p. 17, paragraph 044. The method may also include representing an updated status condition of the floating point operand within the floating point operand. *Id.*

The invention, as recited by independent claim 40, relates to a method of encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register. *Id.* at pp. 16-17, paragraphs 043-044. The method may include receiving a floating point instruction and accessing a floating point operand to be processed as part of processing the floating point instruction. *Id.* The method may also include decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand. *Id.* Further, the method may include encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand. *Id.* at pp. 17-18, paragraphs 044-046.

The invention, as recited by independent claim 44, relates to a computer-readable medium on which is stored a set of instructions for encoding a floating point operand with status information without maintaining the status information in a floating point status register, which when executed perform steps. *Id.* at p. 17, paragraph 044;

p. 22, paragraph 063; p. 18, paragraph 047. The steps may include determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation. *Id.* at p. 17, paragraph 044; p. 18, paragraph 047. The steps may also include representing an updated status condition of the floating point operand within the floating point operand. *Id.*

The invention, as recited by independent claim 51, relates to a computer-readable medium on which is stored a set of instructions for encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, which when executed perform steps. *Id.* at pp. 16-17, paragraphs 043-044; p. 18, paragraph 047. The steps may include receiving a floating point instruction and accessing a floating point operand to be processed as part of processing the floating point instruction. *Id.* The steps may also include decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand. *Id.* Further, the steps may include encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand. *Id.* at pp. 17-18, paragraphs 044-046; p. 18, paragraph 047.

VI. GROUNDS OF REJECTION TO BE REVIEWED

A. Claims 1-5, 21-31, and 33-54 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,995,991 to Huang et al. ("*Huang*"). *Office Action mailed June 6, 2005* at p. 2. Appellant appeals this rejection of those claims.

B. Claims 1-5, 21-31, and 33-54 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,009,551 to Lynch et al. ("*Lynch*"). *Office Action mailed June 6, 2005* at p. 2. Appellant appeals this rejection of those claims.

C. Claims 6-20 and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Huang*. *Office Action mailed June 6, 2005* at p. 2. Appellant appeals this rejection of those claims.

D. Claims 6-20 and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Lynch*. *Office Action mailed June 6, 2005* at p. 2. Appellant appeals this rejection of those claims.

VII. ARGUMENT

A. Introduction

In view of the reasoning set forth below, Appellant respectfully requests the Board to reverse the Examiner's rejections. Each ground of rejection is treated under a separate heading, with claims argued separately placed under a subheading identifying the claim(s) by number.

B. The rejection of claims 1-5, 21-31, and 33-54 under 35 U.S.C. § 102(b) as being anticipated by *Huang*

In order to properly establish that *Huang* anticipates Appellant's claimed invention under 35 U.S.C. § 102, each and every element of each of the claims in issue must be found, either expressly described or under principles of inherency, in that single reference. Furthermore, "[t]he identical invention must be shown in as complete detail as is contained in the ... claim." *See* M.P.E.P. § 2131, quoting *Richardson v. Suzuki Motor Co.*, 868 F.2d 1126, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989).

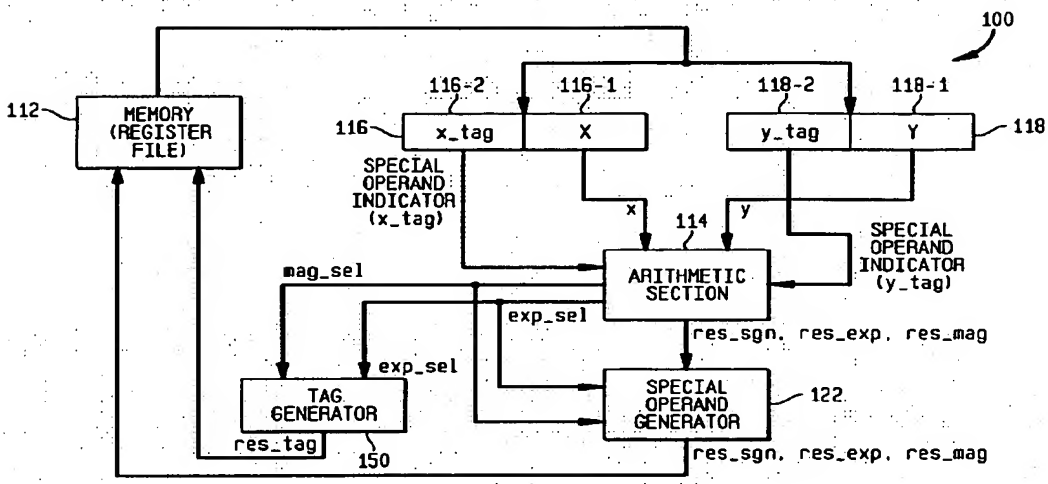
1. Claims 1-5 patentably distinguish from *Huang*

Huang does not disclose each and every element of Appellant's claimed invention. Independent claim 1 calls for a combination including, for example, "a floating point operand data structure" including "embedded status information associated with at least one status condition of the floating point operand data." *Huang* fails to teach or suggest this element.

First, the Examiner asserts that "[t]he Examiner believes that each 'operand register' should have the claimed 'first portion' and 'second portion ... having embedded status' because Hwang et al teach a 'floating point device.'" *Id.* The Examiner's assertion that *Huang* "should have" the elements of claim 1 does not support a rejection under 35 U.S.C. § 102. Further, merely because *Huang* allegedly teaches a "floating point device" does not establish a proper rejection under 35 U.S.C. § 102.

Second, the Examiner's rejection asserts that *Huang* teaches "[e]ach operand register (X or Y) has a first portion (116-1, 118-1) and second portion (116-2, 118-2)." *Office Action mailed June 6, 2005* at p. 4 (emphasis removed). This appears to be an assertion that *Huang's* register 116 (Fig. 4) constitutes the claimed "operand." This is incorrect according to the disclosure of *Huang*. As illustrated in Fig. 4 of *Huang*, which Appellant reproduces below, *Huang* specifically distinguishes between a register and an operand, stating "each of the registers 116 and 118 has an operand . . . portion 116-1 and 118-1." (*Huang,* col. 6, line 66-col. 7, line 1).

FIG. 4



Moreover, "operands X and Y [are] stored in operand portions 116-1 and 118-1 of registers 116 and 118." (*Huang*, col. 7, lines 8-10). Thus, even assuming that tag values x_tag and y_tag correspond to the claimed "status information," (which Appellant does not concede) the Examiner's contention that *Huang's* register 116 containing x_tag 116-2 constitutes the claimed "floating point operand data structure" including "embedded status information associated with at least one status condition of the floating point operand data" is clearly contradicted by *Huang*.

Structures such as those taught by *Huang* were acknowledged in the Background section of Appellant's specification, which states "conditions are typically represented by flags [a form of status information] that are stored in the floating point status register" (paragraph 031). Claim 1 specifically distinguishes over such structures, calling for "a floating point operand data structure" including "embedded status information associated

with at least one status condition of the floating point operand data." *See, e.g.,* Fig. 4, paragraph 061, and paragraph 063 of Appellant's specification.

Accordingly, the Examiner's position that *Huang's* register 116 constitutes an "operand," as recited in the claims, is a clear error. *Huang*, therefore, cannot anticipate independent claim 1 and dependent claims 2-5. Appellant requests that the Board allow these claims.

2. Claim 2 patentably distinguishes from *Huang*

Huang does not disclose each and every element of dependent claim 2. Dependent claim 2 recites determining "at least one status condition" from "the embedded status information without regard to memory storage external to the data structure" (emphasis added). The Examiner has not addressed any of the elements recited by dependent claim 2. *See Office Action mailed November 15, 2004* at p. 2. These elements are not taught by *Huang*. In fact, as discussed above, *Huang* discloses the opposite of the claimed structure, teaching use of registers 116 and 118, which are memory storage external to the operand data structure (*Huang*, FIG. 4; col. 6, line 65). Accordingly, *Huang* cannot anticipate dependent claim 2. Appellant requests that the Board allow this claim.

3. Claim 3 patentably distinguishes from *Huang*

Huang does not disclose each and every element of dependent claim 3. Dependent claim 3 recites "[t]he floating point operand data structure of claim 2, wherein the memory storage external to the data structure is a floating point status register." As with claim 2, the Examiner has not addressed any of the elements recited by dependent claim 3. *See Office Action mailed November 15, 2004* at p. 2. These

elements are not taught by *Huang*. In fact, as discussed above, *Huang* discloses the opposite of the claimed structure, teaching use of registers 116 and 118 (*Huang*, FIG. 4; col. 6, line 65), which is the structure that claim 3 (as depending from claim 2) distinguishes. Accordingly, *Huang* cannot anticipate dependent claim 3. Appellant requests that the Board allow this claim.

4. Claim 4 patentably distinguishes from *Huang*

Huang does not disclose each and every element of dependent claim 4. Dependent claim 4 recites "[t]he enhanced floating point operand data structure of claim 3, wherein the outcome of a conditional floating point instruction is based on the embedded status information without regard to contents of the floating point status register" (emphasis added). Once again, the Examiner has not addressed any of the elements recited by dependent claim 4. *See Office Action mailed November 15, 2004* at p. 2. These elements are not taught by *Huang*. In fact, as discussed above, *Huang* discloses the opposite of the claimed structure, teaching use of the contents of registers 116 and 118 (*Huang*, FIG. 4; col. 6, line 65). Accordingly, *Huang* cannot anticipate dependent claim 4. Appellant requests that the Board allow this claim.

5. Claims 21-31 patentably distinguish from *Huang*

Huang does not disclose each and every element of Appellant's claimed invention. Independent claim 21 calls for a combination including, for example,

> an operand memory storage device for maintaining the floating point operand;
>
> a control unit in communication with the operand memory storage device, the control unit receiving at least one floating point instruction associated with the at least one floating point operation and generating at least one control signal related to the at least one floating point operation; and

> a first functional processing unit in communication with the operand memory storage device and the control unit, the first functional processing unit capable processing the floating point operand and storing status information within the processed floating point operand

(emphasis added). Unsurprisingly, the Examiner has yet again failed to address claim elements such as "operand memory storage device," "control unit," "control signal," and "functional processing unit ... capable [of] ... storing status information within the processed floating point operand" *See Office Action mailed November 15, 2004* at p. 3. *Huang* fails to teach at least these elements of claim 21.

Moreover, *Huang* does not teach or suggest a "functional processing unit ... capable [of] ... storing status information within the processed floating point operand," as recited by independent claim 21 (emphasis added). Even assuming that *Huang's* tag constitutes the claimed "status information," which Appellant does not concede, *Huang* illustrates in FIG. 4 a separate tag generator 150 and separate special operand generators 122. Such teachings do not constitute a teaching or suggestion of a "functional processing unit ... capable [of] ... storing status information within the processed floating point operand," as recited by independent claim 21. As discussed above regarding the rejection of claim 1, *Huang* does not teach or suggest "storing status information within the processed floating point operand," as recited by claim 21.

Independent claim 27, although of different scope, recites elements similar to independent claim 21. Accordingly, for at least the reasons discussed above with respect to independent claim 21, *Huang* cannot anticipate independent claim 27. Claims 22-26 and 28-31 depend from independent claims 21 and 27, respectively, and therefore include all of the elements recited therein. Appellant therefore requests that the Board to allow claims 21-31.

6. Claims 23 and 29 patentably distinguish from *Huang*

Huang does not disclose each and every element of dependent claims 23 and 29. Dependent claim 23 recites "[t]he floating point system of claim 21, wherein the first functional processing unit is capable of embedding the status information related to the processed floating point operand within predetermined fields of the processed floating point operand" (emphasis added). The Examiner has not addressed these elements. *See Office Action mailed November 15, 2004* at p. 3. As discussed above, *Huang* merely teaches storing a tag in a separate tag portion 118-2 (*Huang*, FIG. 4; col. 6, line 66 through col. 7, line 2). Such a teaching by *Huang* constitutes neither the claimed "embedding ... within ... the processed floating point operand," nor the claimed "embedding ... within predetermined fields of the processed floating point operand." Dependent claim 29, although of different scope, recites similar elements to dependent claim 23. Accordingly, *Huang* cannot anticipate dependent claims 23 and 29. Appellant requests that the Board allow these claims.

7. Claims 24 and 30 patentably distinguish from Huang

Huang does not disclose each and every element of dependent claims 24 and 30. Dependent claim 24 recites "[t]he floating point system of claim 21, wherein the first functional processing unit is capable of providing the processed floating point operand to the operand memory storage device without storing the status information to a separate status memory device" (emphasis added). The Examiner has not addressed these elements. *See Office Action mailed November 15, 2004* at p. 3. Indeed, *Huang* clearly illustrates using a separate memory storage device, tag generator 150, to generate tag values which are stored in register 116-2 (*Huang*, FIG. 4; col. 7, lines

31-34). Such a teaching by *Huang* does not constitute the claimed "providing the processed floating point operand to the operand memory storage device without storing the status information to a separate status memory device" (emphasis added). Dependent claim 30, although of different scope, recites elements similar to dependent claim 24. Accordingly, *Huang* cannot anticipate dependent claims 24 and 30. Appellant requests that the Board allow these claims.

8. Claims 25 and 31 patentably distinguishes from Huang

Huang does not disclose each and every element of dependent claims 25 and 31. Dependent claim 25 recites "[t]he floating point system of claim 24, wherein the control unit is operative to condition the outcome of the floating point instruction based upon the status information within the processed floating point operand without accessing the separate status memory device" (emphasis added). The Examiner has not addressed these elements, nor does *Huang* teach or suggest such elements. *See Office Action mailed November 15, 2004* at p. 3. Dependent claim 31, although of different scope, recites elements similar to dependent claim 25. Accordingly, *Huang* cannot anticipate dependent claims 25 and 31. Appellant requests that the Board allow these claims.

9. Claims 26 patentably distinguishes from Huang

Huang does not disclose each and every element of dependent claim 26. Dependent claim 25 recites "[t]he floating point system of claim 21 further comprising a second functional processing unit in communication with the memory storage device and the control unit, the second functional processing unit being capable of processing a second floating point operand and storing status information related to the second

floating point operand while the status information related to the first floating point operand is preserved" (emphasis added). The Examiner has not addressed this element, nor does *Huang* teach or suggest such elements. *See Office Action mailed November 15, 2004* at p. 3.

Moreover, even assuming that *Huang's* arithmetic section 114 (FIG. 4) corresponds to the claimed "functional unit," *Huang* teaches only a single such unit, and the Examiner concedes as much, stating that *Huang* does not teach an "additional function [sic] processing unit." *Office Action mailed November 15, 2004* at p. 4. Accordingly, *Huang* does not teach or suggest a "second functional processing unit ... capable of ... storing second status information related to the floating point operand while the status information related to the first floating point operand is preserved," as recited by dependent claim 26. *Huang*, therefore, cannot anticipate dependent claim 26. Appellant requests that the Board allow this claim.

10. Claims 33-39 and 44-50 patentably distinguish from *Huang*

Huang does not disclose each and every element of Appellant's claimed invention. Independent claim 33 recites

> A method of encoding a floating point operand with status information without maintaining the status information in a floating point status register, comprising:
> determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation; and
> representing an updated status condition of the floating point operand within the floating point operand

(emphasis added).

The Examiner has not addressed the elements of independent claim 33, nor does *Huang* teach or suggest such elements. *See Office Action mailed November 15, 2004* at p. 3.

Moreover, as discussed above regarding claim 1, *Huang* does not teach or suggest "representing an updated status condition of the floating point operand within the floating point operand," as recited by independent claim 33. Accordingly, *Huang* cannot anticipate this claim.

Further, *Huang* clearly does not teach or suggest "encoding a floating point operand with status information without maintaining the status information in a floating point status register," as recited by claim 33. Rather, *Huang* explicitly teaches use of operand registers 116 and 118 that maintain tag values (alleged status information) in a separate portion 116-2 and 118-2. Such teachings do not constitute a teaching or suggest of "encoding a floating point operand with status information without maintaining the status information in a floating point status register," as recited by claim 33. For at least this additional reason, *Huang* cannot anticipate independent claim 33.

Structures such as those taught by *Huang* were acknowledged in the Background section of Appellant's specification, which states "conditions are typically represented by flags [a form of status information] that are stored in the floating point status register" (paragraph 031). Independent claim 33 specifically distinguish over such structures, calling for "encoding a floating point operand with status information without maintaining the status information in a floating point status register" (emphasis added). *See, e.g.,* Fig. 4, paragraph 061, and paragraph 063 of Appellant's specification.

Independent claim 44, although of different scope, recites elements similar to independent claim 33. Claims 34-39 and 45-50 depend from independent claims 33 and 44, respectively, and therefore include all of the elements recited therein. Because *Huang* does not teach or suggest each and every element recited by independent claims 33 and 44 and required by dependent claims 34-39 and 45-50, *Huang* cannot anticipate these claims. Accordingly, Appellant requests that the Board allow claims 33-39 and 44-50.

11. Claims 40-43 and 51-54 patentably distinguish from *Huang*

Huang does not disclose each and every element of Appellant's claimed invention. Independent claim 40 recites

> A method of encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, comprising:
>
> receiving a floating point instruction;
>
> accessing a floating point operand to be processed as part of processing the floating point instruction;
>
> decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand; and
>
> encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand

(emphasis added). As discussed above with respect to independent claim 33, *Huang* fails to teach at least "encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register" and "encoding a resulting status condition ... within the floating point operand," as recited by independent claim 40. Accordingly, *Huang* cannot anticipate independent claim 40.

Moreover, *Huang* does not teach or suggest "decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand," as recited by independent claim 40. Rather, as discussed above, *Huang* receives "status conditions" from tag portion 116-2 (FIG. 4), which is separate from operand portion 116-1. Such teachings of *Huang* do not constitute a teaching or suggestion of "decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand," as recited by independent claim 40. Accordingly, for at least this additional reason, *Huang* cannot anticipate independent claim 40.

Independent claim 51, although of different scope, recites elements similar to claim 40. Claims 41-43 and 52-54 depend from independent claims 40 and 51, respectively, and therefore include all of the elements recited therein. Because *Huang* fails to teach or suggest each and every element recited by independent claims 40 and 51 and required by dependent claims 41-43 and 52-54, *Huang* cannot anticipate these claims. Appellant therefore respectfully requests that the Board allow claims 40-43 and 51-54.

C. The rejection of claims 1-5, 21-31, and 33-54 under 35 U.S.C. § 102(b) as being anticipated by *Lynch*

1. Claims 1-5 patentably distinguish from *Lynch*

Lynch does not disclose each and every element of Appellant's claimed invention. Independent claim 1 calls for a combination including, for example, "a floating point operand data structure" including "embedded status information associated with at least one status condition of the floating point operand data." *Lynch* fails to teach or suggest this element.

First, the rejection does not make clear what portion(s) of *Lynch* allegedly teaches the claimed "status information." *See Office Action mailed November 15, 2004* at p. 3. The Examiner merely asserts that "[t]he examiner believes that 'Register stack' should have the claimed 'first portion' and 'second portion ... having embedded status' because Lynch et al teach a 'floating point device.'" *Office Action mailed June 6, 2005* at p. 4. As discussed above, the Examiner's assertion that *Lynch* "should have" the elements of claim 1 does not support a rejection under 35 U.S.C. § 102. Further, merely because *Lynch* allegedly teaches a "floating point device" does not establish a proper rejection under 35 U.S.C. § 102. These conclusory statements do not constitute a teaching or suggestion of "a floating point operand data structure" including "embedded status information associated with at least one status condition of the floating point operand data," as recited by claim 1 (emphasis added).

Second, the Examiner has not made clear how *Lynch's* "tag values" relate to, for example, "embedded status information," as recited by claim 1. Even assuming that *Lynch's* tag value constitutes "status," (which Appellant does not concede) the tag value of *Lynch* is not included in a "floating point operand data structure" having "embedded status information associated with at least one status condition of the floating point operand data," as recited by claim 1.

Fig. 4 of *Lynch*, which Appellant reproduces below, clearly illustrates that Tag Field 89 (alleged status) and Reg Field 87 (operand) are separate from each other and are stored within register stack 84.

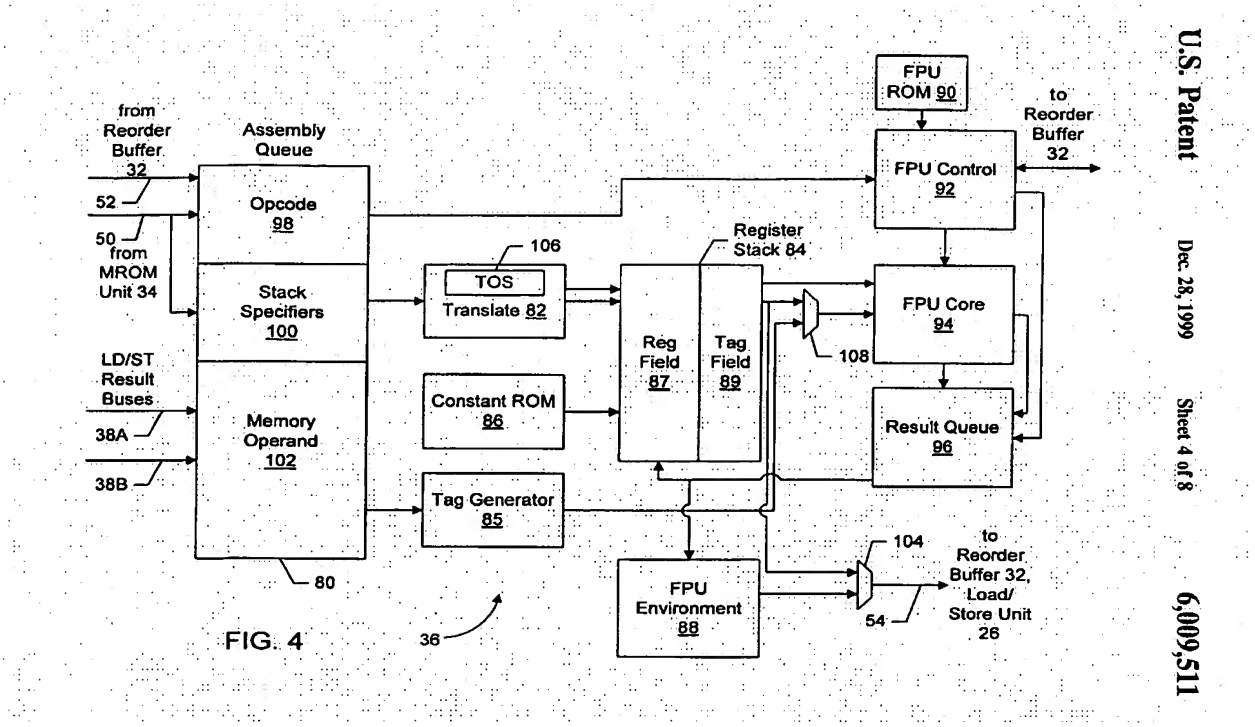


FIG. 4

Moreover, *Lynch* specifically states that element 84 is a register stack, not a floating point operand data structure, and that register stack 84 contains a separate Reg Field 87 for storing an operand and a separate Tag Field 89 for storing a tag (alleged status). *See Lynch*, col. 15, lines 63-67. This does not constitute a teaching or suggestion of "a floating point operand data structure" including "embedded status information associated with at least one status condition of the floating point operand data," as recited by independent claim 1 and required by dependent claims 2-5.

Structures such as those taught by *Lynch* were acknowledged in the Background section of Appellant's specification, which states "conditions are typically represented by flags [a form of status information] that are stored in the floating point status register" (paragraph 031). Claim 1 specifically distinguishes over such structures, calling for "a floating point operand data structure" including "embedded status information associated

with at least one status condition of the floating point operand data" (emphasis added). *See, e.g., Appellant's specification* at Fig. 4, paragraph 061, and paragraph 063.

Because the *Lynch* does not teach or suggest "a floating point operand data structure" including "embedded status information associated with at least one status condition of the floating point operand data," *Lynch* cannot anticipate claims 1-5. Accordingly, the rejection of claims 1-5 under 35 U.S.C. § 102(b) based on *Lynch* is improper. Appellant requests the Board to allow these claims.

2. Claim 2 patentably distinguishes from *Lynch*

Lynch does not disclose each and every element of dependent claim 2. Dependent claim 2 recites determining "at least one status condition" from "the embedded status information without regard to memory storage external to the data structure" (emphasis added). The Examiner has not addressed any of the elements recited by dependent claim 2. *See Office Action mailed November 15, 2004* at p. 3. In fact, these elements are not taught by *Lynch*. Indeed, as discussed above, *Lynch* discloses the opposite of the claimed structure, teaching use of a Register Stack 84 including a separate Tag Field 89 (alleged status), which is memory storage external to *Lynch's* Reg Field 87 (alleged operand) (*Lynch*, FIG. 4). Accordingly, *Lynch* cannot anticipate dependent claim 2. Appellant requests that the Board allow this claim.

3. Claim 3 patentably distinguishes from *Lynch*

Lynch does not disclose each and every element of dependent claim 3. Dependent claim 3 recites "[t]he floating point operand data structure of claim 2, wherein the memory storage external to the data structure is a floating point status register." Once again, the Examiner has not addressed any of the elements recited by

dependent claim 3. *See Office Action mailed November 15, 2004* at p. 3. In fact, *Lynch* does not teach or suggest the elements of claim 3. Indeed, as discussed above, *Lynch* discloses the opposite of the claimed structure, teaching use of a Register Stack 84 including a separate Tag Field 89 (*Lynch*, FIG. 4). Accordingly, *Lynch* cannot anticipate dependent claim 3. Appellant requests that the Board allow this claim.

4. Claim 4 patentably distinguishes from *Lynch*

Lynch does not disclose each and every element of dependent claim 4. Dependent claim 4 recites "[t]he enhanced floating point operand data structure of claim 3, wherein the outcome of a conditional floating point instruction is based on the embedded status information without regard to contents of the floating point status register" (emphasis added). The Examiner has not addressed any of the elements recited by dependent claim 4. *See Office Action mailed November 15, 2004* at p. 3. In fact, *Lynch* does not teach or suggest these elements. Indeed, as discussed above, *Lynch* teaches the opposite of the claimed structure, teaching use of a Register Stack 84 (*Lynch*, FIG. 4). Accordingly, *Lynch* cannot anticipate dependent claim 4. Appellant requests that the Board allow this claim.

5. Claims 21-31 patentably distinguish from *Lynch*

Lynch does not disclose each and every element of Appellants' claimed invention. Independent claim 21 calls for a combination including, for example,

> an operand memory storage device for maintaining the floating point operand;
>
> a control unit in communication with the operand memory storage device, the control unit receiving at least one floating point instruction associated with the at least one floating point operation and generating at least one control signal related to the at least one floating point operation; and

> a first functional processing unit in communication with the operand memory storage device and the control unit, the first functional processing unit capable processing the floating point operand and storing status information within the processed floating point operand

(emphasis added). The Examiner has not addressed how *Lynch* allegedly teaches or suggests the claimed "operand memory storage device," "control unit," "control signal," "functional processing unit ... capable [of] ... storing status information within the processed floating point operand." *See Office Action mailed November 15, 2004* at p. 3. *Lynch* fails to teach at least these elements. Accordingly, *Lynch* cannot anticipate independent claim 21.

Moreover, *Lynch* does not teach or suggest a "functional processing unit ... capable [of] ... storing status information within the processed floating point operand," as recited by independent claim 21. Even assuming that *Lynch's* tag constitutes the claimed "status information," which Appellant does not concede, *Lynch* clearly illustrates in FIG. 4 a separate tag field 89 and a separate register field 87. Such teachings do not constitute a teaching or suggestion of a "functional processing unit ... capable [of] ... storing status information within the processed floating point operand," as recited by independent claim 21. As discussed above regarding the rejection of claim 1 as allegedly anticipated by *Lynch*, *Lynch* does not teach or suggest "storing status information within the processed floating point operand," as recited by claim 21.

Independent claim 27, although of different scope, recites elements similar to independent claim 21. Accordingly, for at least the reasons discussed above with respect to independent claim 21, *Lynch* cannot anticipate independent claim 27. Claims 22-26 and 28-31 depend from independent claims 21 and 27, respectively, and

therefore include all of the elements recited therein. Appellant therefore requests that the Board to allow claims 21-31.

6. Claims 23 and 29 patentably distinguish from *Lynch*

Lynch does not disclose each and every element of dependent claims 23 and 29. Dependent claim 23 recites "[t]he floating point system of claim 21, wherein the first functional processing unit is capable of embedding the status information related to the processed floating point operand within predetermined fields of the processed floating point operand." The Examiner has not addressed these elements. *See Office Action mailed November 15, 2004* at p. 3. As discussed above, *Lynch* merely teaches storing a tag in a separate Tag Field 89 (*Lynch*, FIG. 4). Such a teaching by *Lynch* constitutes neither the claimed "embedding ... within ... the processed floating point operand," nor the claimed "embedding ... within predetermined fields of the processed floating point operand." Dependent claim 29, although of different scope, recites similar elements to dependent claim 23. Accordingly, *Lynch* cannot anticipate dependent claims 23 and 29. Appellant requests that the Board allow these claims.

7. Claims 24 and 30 patentably distinguish from Lynch

Lynch does not disclose each and every element of dependent claims 24 and 30. Dependent claim 24 recites "[t]he floating point system of claim 21, wherein the first functional processing unit is capable of providing the processed floating point operand to the operand memory storage device without storing the status information to a separate status memory device" (emphasis added). The Examiner has not addressed these elements. *See Office Action mailed November 15, 2004* at p. 3. In fact, *Lynch* does not teach or suggest the claimed elements. Indeed, as discussed above, *Lynch*

discloses the opposite of the claimed structure, teaching use of a separate memory storage device, Tag Field 89 (*Lynch*, FIG. 4; col. 5, lines 44-45). Such a teaching by *Lynch* does not constitute the claimed "providing the processed floating point operand to the operand memory storage device without storing the status information to a separate status memory device," as recited by dependent claim 24 (emphasis added). Dependent claim 30, although of different scope, recites elements similar to dependent claim 24. Accordingly, *Lynch* cannot anticipate dependent claims 24 and 30. Appellant requests that the Board allow these claims.

8. Claims 25 and 31 patentably distinguishes from Lynch

Lynch does not disclose each and every element of dependent claims 25 and 31. Dependent claim 25 recites "[t]he floating point system of claim 24, wherein the control unit is operative to condition the outcome of the floating point instruction based upon the status information within the processed floating point operand without accessing the separate status memory device" (emphasis added). The Examiner has not addressed these elements, nor does *Lynch* teach or suggest these elements. *See Office Action mailed November 15, 2004* at p. 3. Dependent claim 31, although of different scope, recites elements similar to dependent claim 25. Accordingly, *Lynch* cannot anticipate dependent claims 25 and 31. Appellant requests that the Board allow these claims.

9. Claims 26 patentably distinguishes from Lynch

Lynch does not disclose each and every element of dependent claim 26. Dependent claim 25 recites "[t]he floating point system of claim 21 further comprising a second functional processing unit in communication with the memory storage device and the control unit, the second functional processing unit being capable of processing

a second floating point operand and storing status information related to the second floating point operand while the status information related to the first floating point operand is preserved," as recited by dependent claim 26 (emphasis added). The Examiner has not addressed these elements, nor does *Lynch* teach or suggest these elements. *See Office Action mailed November 15, 2004* at p. 3.

Moreover, the Examiner concedes that *Lynch* does not teach or suggest an "additional function [sic] processing unit." *Office Action mailed November 15, 2004* at p. 4. Accordingly, *Lynch* cannot teach or suggest a "second functional processing unit ... capable of ... storing second status information related to the floating point operand while the status information related to the first floating point operand is preserved," as recited by dependent claim 26. *Lynch*, therefore, cannot anticipate dependent claim 26. Appellant requests that the Board allow this claim.

10. Claims 33-39 and 44-50 patentably distinguish from *Lynch*

Lynch does not disclose each and every element of Appellant's claimed invention. Independent claim 33 recites

> A method of encoding a floating point operand with status information without maintaining the status information in a floating point status register, comprising:
> determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation; and
> representing an updated status condition of the floating point operand within the floating point operand

(emphasis added).

The Examiner has not addressed the elements of independent claim 33, nor does *Lynch* teach or suggest such elements. *See Office Action mailed November 15, 2004* at p. 3.

Moreover, as discussed above regarding the rejection of claim 1 as anticipated by *Lynch*, *Lynch* does not teach or suggest "representing an updated status condition of the floating point operand within the floating point operand," as recited by independent claim 33. Accordingly, *Lynch* cannot anticipate this claim.

Further, *Lynch* clearly does not teach or suggest "encoding a floating point operand with status information without maintaining the status information in a floating point status register," as recited by claim 33. Rather, *Lynch* explicitly teaches use of Register Stack 89 including a Tag Field 89 (storing alleged status information) separate from Reg Field 87 (storing alleged operand). *Lynch,* FIG. 4, col. 15, lines 63-67, col. 16, lines 11-13). Such teachings do not constitute a teaching or suggest of "encoding a floating point operand with status information without maintaining the status information in a floating point status register," as recited by claim 33. For at least this additional reason, *Lynch* cannot anticipate independent claim 33.

Structures such as those taught by *Lynch* were acknowledged in the Background section of Appellant's specification, which states "conditions are typically represented by flags [a form of status information] that are stored in the floating point status register" (paragraph 031). Independent claim 33 specifically distinguish over such structures, calling for "encoding a floating point operand with status information without maintaining the status information in a floating point status register" (emphasis added). *See, e.g.,* Fig. 4, paragraph 061, and paragraph 063 of Appellant's specification.

Independent claim 44, although of different scope, recites elements similar to independent claim 33. Claims 34-39 and 45-50 depend from independent claims 33 and 44, respectively, and therefore include all of the elements recited therein. Because

Lynch does not teach or suggest each and every element recited by independent claims 33 and 44 and required by dependent claims 34-39 and 45-50, *Lynch* cannot anticipate these claims. Accordingly, Appellant requests that the Board allow claims 33-39 and 44-50.

11. Claims 40-43 and 51-54 patentably distinguish from *Lynch*

Lynch does not disclose each and every element of Appellant's claimed invention. Independent claim 40 calls for a combination including, for example,

> A method of encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, comprising:
>
> receiving a floating point instruction;
>
> accessing a floating point operand to be processed as part of processing the floating point instruction;
>
> decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand; and
>
> encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand

(emphasis added). As discussed above with respect to independent claim 33, *Lynch* fails to teach at least "encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register" and "encoding a resulting status condition ... within the floating point operand," as recited by independent claim 40. Accordingly, *Lynch* cannot anticipate independent claim 40.

Moreover, *Lynch* does not teach or suggest "decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand," as recited by independent claim 40. Rather, as discussed above, *Lynch*

allegedly receives "status conditions" from Tag Field 89 (FIG. 4), which is separate from Reg. Field 87. Such teachings of *Lynch* do not constitute a teaching or suggestion of "decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand," as recited by independent claim 40. Accordingly, for at least this additional reason, *Lynch* cannot anticipate independent claim 40.

Independent claim 51, although of different scope, recites elements similar to claim 40. Claims 41-43 and 52-54 depend from independent claims 40 and 51, respectively, and therefore include all of the elements recited therein. Because *Lynch* fails to teach or suggest each and every element recited by independent claims 40 and 51 and required by dependent claims 41-43 and 52-54, *Lynch* cannot anticipate these claims. Appellant therefore respectfully requests that the Board allow claims 40-43 and 51-54.

D. The rejection of claims 6-20 and 32 under 35 U.S.C. § 103(a) as being unpatentable over *Huang*

Several basic factual inquires must be made in order to determine the obviousness or non-obviousness of claims of a patent application under 35 U.S.C. § 103. These factual inquiries, set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 17, 148 USPQ 459, 467 (1966), require the Examiner to:

> (1) Determine the scope and content of the prior art;
> (2) Ascertain the differences between the prior art and the claims in issue;
> (3) Resolve the level of ordinary skill in the pertinent art; and
> (4) Evaluate evidence of secondary considerations.

The obviousness or nonobviousness of the claimed invention is then evaluated in view of the results of these inquiries. *Graham*, 383 U.S. at 17-18, 148 USPQ 467.

Thus, in order to carry the initial burden of establishing a *prima facie* case of obviousness that satisfies the *Graham* standard, the Examiner must show that the prior art reference teaches or suggests all the claim elements. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). The Examiner must also show that there is some suggestion or motivation, either in the reference or in the knowledge generally available to one of ordinary skill in the art, to modify the reference. *In re Rouffet*, 149 F.3d 1350, 47 USPQ2d 1453 (Fed. Cir. 1998). "Even when obviousness is based on a single prior art reference, there must be a showing of a suggestion or motivation to modify the teachings of that reference." *In re Kotzab*, 217 F.3d 1365, 1370, 55 USPQ2d 1313, 1316-17 (Fed. Cir. 1998) (citations omitted). In addition, the suggestion or motivation "must be found in the prior art reference, not in the Appellant's disclosure." *In re Vaeck*, 947 F.2d 488, 493, 20 USPQ2d 1438, 1442 (Fed. Cir. 1991).

1. Claims 6-15 patentably distinguish from *Huang*

Claims 6-15 depend from independent claim 1 and therefore include all of the elements recited therein. As discussed above, *Huang* does not teach each and every element of claim 1. Moreover, the Examiner has not addressed any of the elements of claims 6-15, other than to make a general conclusion that "the features are well known" (*Office Action mailed November 15, 2004* at p. 4). The Examiner has not provided any documentary evidence supporting these general conclusions, despite Appellant's request (*Request for Reconsideration filed August 4, 2005* at pp. 14-15). Because the Examiner has not set forth how *Huang* allegedly teaches or suggests each and every element, has not provided any other reference to cure *Huang's* deficiencies, and has not provided the requisite motivation to modify *Huang* to arrive at Appellant's claimed

invention, no *prima facie* case of obviousness has been established with respect to claims 6-15. Appellant requests the Board to allow these claims.

2. Claims 16-20 patentably distinguishes from *Huang*

No *prima facie* case of obviousness has been established with respect to independent claims 16-20 because *Huang* fails to teach or suggest each and every element recited by the claim and because there is no motivation to modify *Huang* to arrive at Appellant's claimed invention.

Independent claim 16 recites a combination including, for example,

> a first data field having sign information associated with the floating point operand;
>
> a second data field having exponent information associated with the floating point operand; and
>
> a third data field having fractional information associated with the floating point operand, wherein at least one of the first data field, the second data field and the third data field further includes embedded status information associated with at least one operand status condition.

The Examiner asserts that "the examiner believes that a 'floating point data' should have 'sign', 'exponent' and 'fraction' information'" (*Office Action mailed June 6, 2005* at p. 4). As discussed above, the Examiner's "belief" that *Huang* "should have" the elements of claim 16 and that "Huang et al's device is a floating point device" does not establish that *Huang* teaches each and every element of claim 16. The Examiner has not cited any portion of *Huang* as teaching or suggesting these claimed elements.

Moreover, *Huang* does not teach or suggest at least "a first data field having sign information," "a second data field having exponent information," and "a third data field having fractional information," as recited by claim 16. Further, as discussed above, *Huang* fails to teach or suggest at least "embedded status information," (emphasis

added) as recited by independent claim 16. The Examiner does not cite any other reference to cure these deficiencies. Thus, *Huang* does not teach or suggest each and every element recited by independent claim 16 and required by dependent claims 17-20.

Moreover, there is no motivation to modify *Huang* to arrive at Appellant's claimed invention. The threshold for establishing a motivation or suggestion to modify a prior art reference is high. The Federal Circuit has clearly stated that the evidence of a motivation or suggestion to modify a reference must be "clear and particular." *In re Dembicziak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999). Further, the Examiner can satisfy the burden of establishing a *prima facie* case of obviousness "only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to [modify or] combine the relevant teachings of the references." *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988) (citations omitted) (emphasis added). The Federal Circuit has reaffirmed the Examiner's high burden to establish a *prima facie* case of obviousness and has emphasized the requirement of specificity. *See Kotzab*, 217 F.3d at 1370, 55 USPQ2d, at 1317; *see also In re Sang-Su Lee*, 277 F.3d 1338, 61 USPQ2d 1430 (Fed. Cir. 2002).

The Examiner relies on a single reference, *Huang*, in rejecting claim 16. However, the Examiner has not provided any motivation for one of ordinary skill in the art to modify *Huang* to achieve the claimed combination. *See Office Action mailed November 15, 2004* at p. 4. "Even when obviousness is based on a single prior art reference, there must be a showing of a suggestion or motivation to modify the

teachings of that reference." *In re Kotzab*, 217 F.3d 1365, 1370, 55 USPQ2d 1313, 1316-17 (Fed. Cir. 1998) (citations omitted).

Since *Huang* fails to teach each and every claim element, and there is no motivation to modify *Huang* to arrive at Appellant's invention as recited by claim 16 and required by dependent claims 17-20, no *prima facia* case of obviousness has been established for these claims. Appellant requests the Board to allow claims 16-20.

3. Claims 17 and 18 patentably distinguish from *Huang*

Huang does not disclose each and every element of dependent claims 17 and 18, nor is there any motivation to modify *Huang* to arrive at Appellant's claimed invention. As discussed above with regard to claims 3 and 4, the Examiner has not addressed any of the elements recited by dependent claims 17 and 18. *See Office Action mailed November 15, 2004* at p. 4. Accordingly, no *prima facie* case of obviousness has been established for these claims. Appellant requests that the Board allow these claims.

4. Claim 32 patentably distinguishes from *Huang*

Huang does not disclose each and every element of dependent claim 32, nor is there any motivation to modify *Huang* to arrive at Appellant's claimed invention. Claim 32 recites

> The floating point system of claim 31 further comprising an additional functional processing unit in communication with the operand memory register and the control unit, the additional functional processing unit being capable of concurrently processing an additional floating point operand and storing status information related to the additional floating point operand within the additional floating point operand while the status information related to the other floating point operand is preserved within the other floating point operand

(emphasis added). The Examiner merely asserts "[t]he feature is obvious to a person having ordinary skill in the art." *See Office Action mailed November 15, 2004* at p. 4. This bare assertion cannot establish that *Huang* teaches or suggests all elements recited by claim 32. Moreover, the Examiner has not provided any motivation to modify *Huang* to arrive at Appellant's claimed invention. Accordingly, no *prima facie* case of obviousness has been established for claim 32. Appellant requests that the Board allow claim 32.

E. The rejection of claims 6-20 and 32 under 35 U.S.C. § 103(a) as being unpatentable over *Lynch*

1. Claims 6-15 patentably distinguish from *Lynch*

Claims 6-15 depend from independent claim 1 and therefore include all of the elements recited therein. As established above, *Lynch* fails to teach or suggest each and every element of claim 1. The Examiner does not cite any other references to cure the deficiencies of *Lynch* discussed above. Moreover, the Examiner has not addressed any of the elements of claims 6-15, other than to make a general conclusion that "the features are well known" (*Office Action mailed November 15, 2004* at p. 4) and has failed to cite any documentary evidence supporting these general conclusions, despite Appellant's request (*Request for Reconsideration filed August 4, 2005* at p. 21). Because the Examiner has not set forth how *Lynch* allegedly teaches or suggests each and every element, has not provided any other reference to cure *Lynch's* deficiencies, and has not provided the requisite motivation to modify *Lynch* to arrive at Appellant's claimed invention, no *prima facie* case of obviousness has been established with respect to claims 6-15. Appellant requests the Board to allow these claims.

2. Claims 16-20 patentably distinguishes from *Lynch*

No *prima facie* case of obviousness has been established with respect to independent claims 16-20 because *Lynch* fails to teach or suggest each and every element recited by the claim and because there is no motivation to modify *Lynch* to arrive at Appellant's claimed invention.

Independent claim 16 recites a combination including, for example,

> a first data field having sign information associated with the floating point operand;
>
> a second data field having exponent information associated with the floating point operand; and
>
> a third data field having fractional information associated with the floating point operand, wherein at least one of the first data field, the second data field and the third data field further includes embedded status information associated with at least one operand status condition.

The Examiner asserts that "the examiner believes that a 'floating point data' should have 'sign', 'exponent' and 'fraction' information'" (*Office Action mailed June 6, 2005* at p. 4). As discussed above, the Examiner's "belief" that *Lynch* "should have" the elements of claim 16 and that "Lynch et al's device is a floating point device" cannot be substituted for a teaching or suggestion of these claim elements in the prior art. The Examiner has not cited any portion of *Lynch* as teaching or suggesting these features.

Moreover, *Lynch* does not teach or suggest at least "a first data field having sign information," "a second data field having exponent information," and "a third data field having fractional information," as recited by claim 16. Further, as discussed above, *Lynch* fails to teach or suggest at least "embedded status information," (emphasis added) as recited by independent claim 16. The Examiner does not cite any other reference to cure these deficiencies.

The Examiner relies on a single reference, *Lynch*, in rejecting claim 16. As noted above, even when obviousness is based on a single prior art reference, there must be a showing of a suggestion or motivation to modify the teachings of that reference. The Examiner has not provided any motivation to modify *Lynch*. *See Office Action mailed November 15, 2004* at p. 4.

Because *Lynch* does not teach or suggest each and every element recited by independent claim 16 and required by dependent claims 17-20, and because there is no motivation to modify *Lynch* to arrive at Appellant's invention as recited by claim 16 and required by dependent claims 17-20, no *prima facia* case of obviousness has been established for these claims. Appellant requests the Board to allow claims 16-20.

3. Claims 17, 18, and 20 patentably distinguish from *Lynch*

Lynch does not disclose each and every element of dependent claims 17, 18, and 20, which are dependent from claim 16, nor is there any motivation to modify *Lynch* to arrive at Appellant's claimed invention. The Examiner has not addressed any of the elements recited by dependent claims 17, 18, and 20. *See Office Action mailed November 15, 2004* at p. 4. Accordingly, no *prima facie* case of obviousness has been established for these claims. Appellant requests that the Board allow these claims.

4. Claim 32 patentably distinguishes from *Lynch*

Lynch does not disclose each and every element of dependent claim 32, nor is there any motivation to modify *Lynch* to arrive at Appellant's claimed invention. Claim 32 recites

> The floating point system of claim 31 further comprising an additional functional processing unit in communication with the operand memory register and the control unit, the additional functional processing unit being capable of concurrently processing an additional floating point

> operand and storing status information related to the additional floating point operand within the additional floating point operand <u>while the status information related to the other floating point operand is preserved within the other floating point operand</u>

(emphasis added). The Examiner merely asserts "[t]he feature is obvious to a person having ordinary skill in the art." *See Office Action mailed November 15, 2004* at p. 4. Once again, the Examiner's bare assertion cannot be substituted for the required teaching or suggestion in the prior art of the elements recited by claim 32, and the required motivation to modify *Lynch* to arrive at Appellant's claimed invention. Accordingly, no *prima facie* case of obviousness has been established for these claims. Appellant requests that the Board allow these claims.

VIII. CONCLUSION

The Examiner has established neither anticipation nor a *prima facie* case of obviousness with respect to the presently appealed claims. For the reasons given above, pending claims 1-54 are allowable and reversal of the Examiner's rejections are respectfully requested.

To the extent any extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Appeal Brief, such extension is hereby respectfully requested. If there are any fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: April 7, 2006

By: _____
Nathan A. Sloan
Reg. No. 56,249

IX. Claims Appendix to Appeal Brief Under Rule 41.37(c)(1)(viii)

1. A floating point operand data structure used in floating point computations and processing within a processing device, comprising:

a first portion of the data structure having floating point operand data; and

a second portion of the data structure having embedded status information associated with at least one status condition of the floating point operand data.

2. The floating point operand data structure of claim 1, wherein the at least one status condition is determined from the embedded status information without regard to memory storage external to the data structure.

3. The floating point operand data structure of claim 2, wherein the memory storage external to the data structure is a floating point status register.

4. The enhanced floating point operand data structure of claim 3, wherein the outcome of a conditional floating point instruction is based on the embedded status information without regard to contents of the floating point status register.

5. The floating point operand data structure of claim 1, wherein the second portion of the data structure comprises at least one bit that is indicative of the at least one status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

6. The floating point operand data structure of claim 5, wherein the overflow status represents one in a group of a +OV status and a -OV status.

7. The floating point operand data structure of claim 6, wherein the overflow status is represented as a predetermined non-infinity numerical value.

8. The floating point operand data structure of claim 5, wherein the underflow status represents one in a group of a +UV status and a -UV status.

9. The floating point operand data structure of claim 5, wherein the underflow status is represented as a predetermined non-zero numerical value.

10. The floating point operand data structure of claim 5, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

11. The floating point operand data structure of claim 10, wherein the second portion of the data structure comprises a plurality of bits indicative of a predetermined type of operand condition resulting in the NaN status.

12. The floating point operand data structure of claim 10, wherein addition, multiplication, maximum and minimum floating point operations on the data structure are commutative.

13. The floating point operand data structure of claim 5, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

14. The floating point operand data structure of claim 1, wherein the second portion of the data structure comprises a plurality of bits indicative of a predetermined type of operand condition resulting in the infinity status.

15. The floating point operand data structure of claim 1, wherein the at least one status condition is associated with at least one floating point operation that generated the enhanced floating point operand data structure.

16. A floating point operand data structure used by a processing device when performing floating point operations, the data structure comprising:

a first data field having sign information associated with the floating point operand;

a second data field having exponent information associated with the floating point operand; and

a third data field having fractional information associated with the floating point operand, wherein at least one of the first data field, the second data field and the

third data field further includes embedded status information associated with at least one operand status condition.

17. The floating point operand data structure of claim 16, wherein the at least one operand status condition is determined from the embedded status information without regard to a floating point status register that is separate from an operand memory storage device for maintaining the floating point operand data structure.

18. The floating point operand data structure of claim 17, wherein the outcome of a conditional floating point instruction is based on the embedded status information without regard to contents of the floating point status register.

19. The floating point operand data structure of claim 16, wherein the embedded status information comprises at least one bit that is indicative of the at least one operand status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

20. The floating point operand data structure of claim 16, wherein the at least one operand status condition is indicative of at least one floating point operation that generated the floating point operand data structure.

21. A floating point system associated with a processing device for performing at least one floating point operation on a floating point operand, the system comprising:

an operand memory storage device for maintaining the floating point operand;

a control unit in communication with the operand memory storage device, the control unit receiving at least one floating point instruction associated with the at least one floating point operation and generating at least one control signal related to the at least one floating point operation; and

a first functional processing unit in communication with the operand memory storage device and the control unit, the first functional processing unit capable processing the floating point operand and storing status information within the processed floating point operand.

22. The floating point system of claim 21, wherein the status information comprises at least one bit that is indicative of an operand status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

23. The floating point system of claim 21, wherein the first functional processing unit is capable of embedding the status information related to the processed floating point operand within predetermined fields of the processed floating point operand.

24. The floating point system of claim 21, wherein the first functional processing unit is capable of providing the processed floating point operand to the operand memory storage device without storing the status information to a separate status memory device.

25. The floating point system of claim 24, wherein the control unit is operative to condition the outcome of the floating point instruction based upon the status information within the processed floating point operand without accessing the separate status memory device.

26. The floating point system of claim 21 further comprising a second functional processing unit in communication with the memory storage device and the control unit, the second functional processing unit being capable of processing a second floating point operand and storing status information related to the second floating point operand while the status information related to the first floating point operand is preserved.

27. A floating point processing system for performing at least one floating point operation on a floating point operand, the system comprising:

an operand memory register for maintaining the floating point operand; and

a functional processing unit in communication with the operand memory register, the functional processing unit being operative to:

receive the floating point operand from the operand memory register,

process the floating point operand to determine status information related to the processed floating point operand, and

embed the status information within the processed floating point operand.

28. The floating point system of claim 27, wherein the status information comprises at least one bit that is indicative of an operand status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

29. The floating point system of claim 27, wherein the functional processing unit is further operable to embed the status information within at least one predetermined field of the processed floating point operand.

30. The floating point system of claim 27, wherein the functional processing unit is capable of storing the processed floating point operand in the operand memory register without storing the status information in a floating point status register separate from the operand memory register.

31. The floating point system of claim 27 further comprising a control unit in communication with the operand memory register and the functional processing unit,

the control unit being operative to condition the outcome of the floating point instruction based only upon the status information within the processed floating point operand.

32. The floating point system of claim 31 further comprising an additional functional processing unit in communication with the operand memory register and the control unit, the additional functional processing unit being capable of concurrently processing an additional floating point operand and storing status information related to the additional floating point operand within the additional floating point operand while the status information related to the other floating point operand is preserved within the other floating point operand.

33. A method of encoding a floating point operand with status information without maintaining the status information in a floating point status register, comprising:

determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation; and

representing an updated status condition of the floating point operand within the floating point operand.

34. The method of claim 33, wherein the determining step further comprises identifying the status condition and the updated status condition from only embedded status information within the floating point operand.

35. The method of claim 33, wherein the status condition and the updated status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

36. The method of claim 33, wherein the representing step further comprises embedding updated status information within the floating point operand after execution of the floating point operation, the updated status information representing the updated status condition.

37. The method of claim 33, wherein the updated status condition is indicative of a previous floating point operation that resulted in the floating point operand.

38. The method of claim 33 further comprising conditioning a subsequent floating point operation based only upon the updated status information within the floating point operand.

39. The method of claim 33 further comprising processing an additional floating point operand and representing updated status information related to the additional floating point operand within the additional floating point operand while the updated status information related to the other floating point operand is preserved.

40. A method of encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, comprising:

receiving a floating point instruction;

accessing a floating point operand to be processed as part of processing the floating point instruction;

decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand; and

encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand.

41. The method of claim 40 further comprising conditioning execution of a subsequent floating point instruction based only on the updated status information within the floating point operand.

42. The method of claim 40, wherein the initial status condition and resulting status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

43. The method of claim 40 further comprising processing an additional floating point operand and encoding an updated status information into the additional

floating point operand while the updated status information related to the other floating point operand is preserved.

44. A computer-readable medium on which is stored a set of instructions for encoding a floating point operand with status information without maintaining the status information in a floating point status register, which when executed perform steps comprising:

determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation; and

representing an updated status condition of the floating point operand within the floating point operand.

45. The computer-readable medium of claim 44, wherein the determining step further comprises identifying the status condition and the updated status condition from only embedded status information within the floating point operand.

46. The computer-readable medium of claim 44, wherein the status condition and the updated status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

47. The computer-readable medium of claim 44, wherein the representing step further comprises embedding updated status information within the floating point

operand after execution of the floating point operation, the updated status information representing the updated status condition.

48. The computer-readable medium of claim 44, wherein the updated status condition is indicative of a previous floating point operation that resulted in the floating point operand.

49. The computer-readable medium of claim 44 further comprising conditioning a subsequent floating point operation based only upon the updated status information within the floating point operand.

50. The computer-readable medium of claim 44 further comprising processing an additional floating point operand and representing updated status information related to the additional floating point operand within the additional floating point operand while the updated status information related to the other floating point operand is preserved.

51. A computer-readable medium on which is stored a set of instructions for encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, which when executed perform steps comprising:

receiving a floating point instruction;

accessing a floating point operand to be processed as part of processing the floating point instruction;

decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand; and

encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand.

52. The computer-readable medium of claim 51 further comprising conditioning execution of a subsequent floating point instruction based only on the updated status information within the floating point operand.

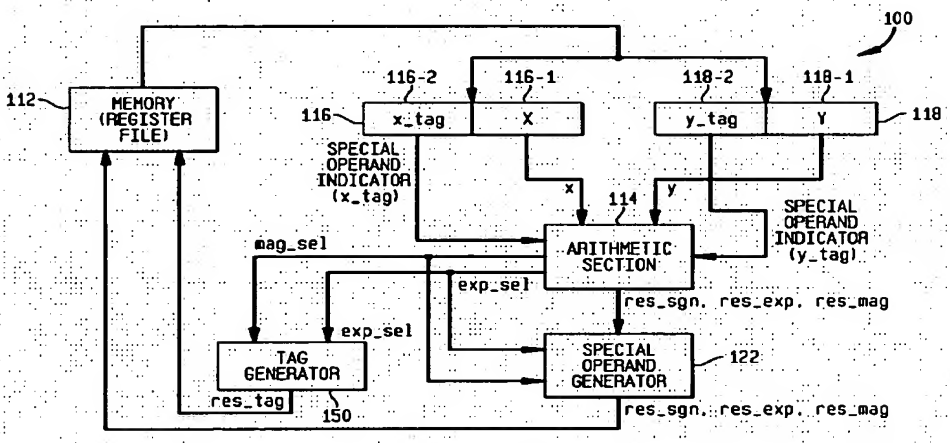
53. The computer-readable medium of claim 51, wherein the initial status condition and resulting status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

54. The computer-readable medium of claim 51 further comprising processing an additional floating point operand and encoding an updated status information into the additional floating point operand while the updated status information related to the other floating point operand is preserved.

X. Evidence Appendix to Appeal Brief Under Rule 41.37(c)(1)(ix)

Appellant relies on Fig. 4 of *Huang* and on Fig. 4 of *Lynch*, both of which are reproduced below.

FIG. 4



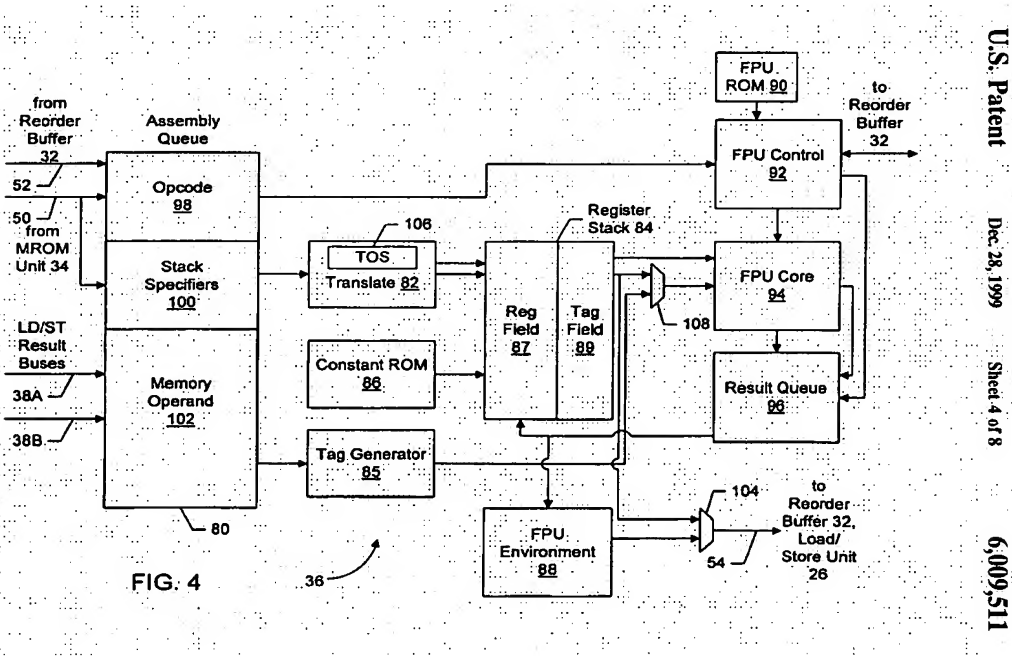


FIG. 4

U.S. Patent Dec. 28, 1999 Sheet 4 of 8 6,009,511

XI. Related Proceedings Appendix to Appeal Brief Under Rule 41.37(c)(1)(x)

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,579, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,595, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,584, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,587, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,647, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,580, in which an Appeal Brief was filed concurrently herewith.